

# Cahier des charges

## Description de notre projet

Création d'une version remasterisé du jeu Frogger 1 et 2. Le nom sera Crossy Frog.

## Nom Equipe

Satari

## Présentation des Membres de l'équipe

Grégoire Dolidon : **Graphiste**

À l'aise avec les outils de création artistique, son rôle principal est la réalisation de tous les graphismes du jeu.

Hakim Izm : **Développeur et chargé de production**

Polyvalent dans tous les domaines et en particulier en orthographe, il est le développeur secondaire et relecteur/correcteur de tous les documents.

Louis Huort : **Développeur (chef de projet)**

Bilingue Qt, il est le développeur principal, s'occupant ainsi du codage et des différents diagrammes.

Mathis Guessard : **Chargé de production**

Multitâche, chargé de la rédaction des comptes rendu et des livrables, ainsi que du suivi du projet.

## Communication

Nous utilisons la plateforme *Discord* sur laquelle nous avons créé un serveur de travail qui nous permet de discuter par messages textuels, d'organiser des réunions en appel vocal, de partager des documents et des informations de manière très rapide.

## Stockage partage de documents et travail collaboratif

La *forge GitLab* de l'IUT nous permet d'héberger tous les fichiers relatifs à notre projet (code, diagrammes, compte-rendu) et de pouvoir travailler de manière collaborative. Ainsi, toutes les personnes du groupe ont le rôle de "maintainer".

*Figma* est le site sur lequel nous avons réalisé les maquettes visuelles de l'interface utilisateur de notre jeu. Toutes ces maquettes sont enregistrées dans un fichier que nous nous partageons.

Nous utilisons *Drawio* pour la réalisation des différents diagrammes afférant à notre jeu. Ces diagrammes sont principalement modifiés par Louis et Hakim avec l'accord de l'équipe.

## Pilotage suivi/avancement projet

Pour la progression de notre projet, nous utilisons principalement Miro qui nous permet de mettre en commun des notes et des idées de manière très visuelle. Cette plateforme nous a beaucoup aidé dans l'élaboration de notre cahier des charges, des comptes-rendus de nos réunions, de la répartition des rôles et de la description de nos objectifs.

Nous avons aussi utilisé le logiciel Gantt Project pour la mise en place de notre calendrier prévisionnel qui nous permettra de suivre des objectifs précis et de respecter les délais prévus.

## Contraintes techniques

Pour le développement de notre jeu, nous utilisons l'IDE Qt Creator qui est configuré pour pouvoir automatiquement se synchroniser avec le GitLab de l'IUT pour permettre un travail collaboratif plus confortable et un suivi des mises à jour.

## Livrables

- Jeu fonctionnel
- Diagrammes de conception
- Support de la soutenance orale
- Documentation afférant à notre jeu

## Cahier des charges V1 :

- Jeu :

Détails techniques :

- Frames par seconde : 60 (60Hz = 16ms)
- Taille de l'écran : 17\*15 cases
- Taille d'une case : 52 pixels

### 1. Mode de jeu :

Crossy Frog V1 n'aura qu'un seul mode de jeu fonctionnel, c'est un mode de jeu avec une carte sans fin (endless) où le joueur devra réussir à aller le plus loin possible, la vitesse de défilement de la carte augmente au fil du temps. Le score correspond au nombre de mètres parcourus (1 case = 1 mètre). La grenouille ne doit pas toucher les blocs d'eau ou les obstacles (comme les voitures).

Détails techniques :

- Vitesse du jeu : 10% - 99%
- Augmentation de la vitesse du jeu : +1% toutes les 12secondes
- Score en mètres, +1 à chaque bloc de haut en plus
- Si la grenouille est sur un bloc infranchissable (eau, voiture, etc.), la partie s'arrête. Détection des positions à chaque tour de boucle.
- L'affichage des patterns se fait à l'aide de 3 patterns comme vus sur les maquettes, dès qu'un pattern sort de l'écran, il retourne en haut et la disposition des blocs est changée.

### 2. Choisir les personnages :

Dans n'importe quel mode de jeu, les personnages pourront être choisis parmi une liste de grenouilles réalisées par notre équipe. Le personnage au centre sera le personnage actuellement sélectionné et les autres seront disposés sur le côté, on naviguera entre eux de gauche à droite.

Détails techniques :

- Les grenouilles seront faites au format 32\*32 pixels
- Elles seront redimensionnées à la taille des cases du jeu soit 52\*52 pixels
- La navigation se fera à l'aide des touches ZQSD, des touches directionnelles et Entrée

### 3. Écran de chargement :

Le jeu disposera d'un écran de chargement à son lancement car les images sont nombreuses et peuvent prendre du temps à charger lors du lancement du jeu.

Détails techniques :

- L'écran de chargement s'affichera avec la progression du chargement des items (progress bar)
- Lancement du menu après le chargement complet des items.

### 4. Menu du jeu :

Le jeu disposera d'un menu contenant les éléments suivants : Play/Hall Of Fame/Settings/Exit. Chacun de ces menus donnera accès au widget correspondant.

Détails techniques :

- Tout se fera à l'aide d'un QPainter : l'option du menu en cours sera surlignée d'un carré blanc
- La navigation se fait à l'aide des touches ZQSD ou des touches directionnelles et de la touche entrée pour valider la sélection.

### 5. Pattern :

Les patterns sont composés de cases de taille fixe, ces cases sont en fait des blocs avec une image et des propriétés correspondantes.

Détails techniques :

- Le pattern fait une taille de 17\*7 cases (donc la moitié de l'écran)
- Il y a donc 3 patterns au total pour avoir un jeu fluide

### 6. Blocs :

Les blocs ont comme attribut une image et des propriétés comme (s'il est traversable, s'il donne de la vitesse, etc.)

Détails techniques :

- L'image du bloc est celle qui a été chargée au lancement du jeu pour éviter les lags lors de la génération.

### 7. Grenouille :

La grenouille correspond à l'image choisie par le joueur avant la partie, elle a des attributs de position et une image.

Détails techniques :

- La grenouille correspond à l'image chargée au lancement du jeu
- Son changement de position se fait à l'aide des touches ZQSD ou des touches directionnelles et des QPressEvent associés, elle bouge à chaque mouvement de la taille d'un bloc soit : 52pixels
- À chaque mouvement, le jeu vérifie si celui-ci est possible, par exemple que ça ne ferait pas sortir la grenouille de la carte
- Si la grenouille est en dehors de la carte, alors la partie d'arrête

8. Scores :

Les scores pourront, si l'utilisateur le souhaite, être enregistré à chaque fin de partie et, s'ils font partie des meilleurs, se retrouver dans le hall of fame.

Détails techniques :

- Le score sera enregistré dans un fichier local.
- Ce fichier sera une base de données avec l'extension ".db".

9. Graphismes :

Tous les graphismes du jeu seront réalisés par notre équipe.

Détails techniques :

- Le format des images différera mais respectera toujours les tailles suivantes : 16x16 ou 32x32 ou 64x64 ou 128x128 (unité : pixel).